Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in this application.

Listing of Claims:

Claims 1-139 (Canceled)

Claim 140 (Currently Amended): A memory for use with a hand-held system for playing video games by displaying graphical information based at least in part on user interaction provided through operation of at least one user-manipulable control, the hand-held system including an object attribute memory (OAM) in a memory space from 07000000h to 070003FFh ~~storage~~, ~~a~~ color palette ~~random access memory~~ storage in a memory space from 05000000h to 050003FFh and ~~a~~ video ~~random access memory~~ storage in a memory space from 06000000h to 06017FFFh, the memory storing executable instructions for:

(a) storing no more than ~~writing up to~~ 128 different 48-bit moving object definitions in ~~to~~ the OAM at any one time ~~object attribute memory (OAM) storage~~;

(b) storing no more than ~~writing up to~~ 512 different 15-bit color values in ~~to~~ the color palette ~~random access memory~~ storage at any one time;

(c) writing ~~at least one of~~ moving object data and background data to the video ~~random access memory~~ storage; and

(d) generating a video game display that is responsive to the user-manipulable control and is based at least in part on the contents of the OAM ~~object attribute memory storage~~, the color palette ~~random access memory~~ storage, and the video ~~random access memory~~ storage,

wherein background data is selectively written to the video ~~random access memory~~ storage as either character data or bitmap ~~pixel~~ data so that backgrounds of the video game display are selectively rendered in a character mode or in a bitmap mode.

Claim 141 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for accessing a predetermined storage ~~16-bit memory~~ location, wherein

bit position 0 of the storage ~~memory~~ location specifies vertical blanking interval status;

bit position 1 of the storage ~~memory~~ location specifies horizontal blanking interval status;

bit position 2 of the storage ~~memory~~ location specifies vertical counter matching or non-matching;

bit position 3 of the storage ~~memory~~ location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage ~~memory~~ location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage ~~memory~~ location specifies whether vertical counter matching interrupts are enabled.

Claim 142 (Currently Amended): The memory of claim 141, wherein the predetermined storage ~~memory~~ location is at address 04000004h ~~004h~~.

Claim 143 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for accessing a predetermined storage ~~16-bit memory~~ location, wherein

bit position 3 of the storage ~~memory~~ location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage ~~memory~~ location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage ~~memory~~ location specifies whether vertical counter matching interrupts are enabled.

Claim 144 (Currently Amended): The memory of claim 143, wherein the predetermined storage ~~memory~~ location is at address 04000004h ~~004h~~.

Claim 145 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for accessing a predetermined storage ~~16-bit memory~~ location, wherein

bit positions 0-2 of the storage ~~memory~~ location specify a background mode;

bit position 4 of the storage ~~memory~~ location specifies a display frame selector for selecting between two different frame buffers;

bit position 5 of the storage ~~memory~~ location specifies whether to render objects during horizontal blanking intervals;

bit position 6 of the storage ~~memory~~ location specifies a control bit for selecting between one-dimensional and two-dimensional object character mapping;

bit positions 8-12 of the storage ~~memory~~ location specifies display of four different background screens and display of moving objects;

bit positions 13-14 of the storage ~~memory~~ location select display of two different windows; and

bit position 15 of the storage ~~memory~~ location selects display of an object window.

Claim 146 (Currently Amended): The memory of claim 145, wherein the predetermined storage ~~memory~~ location is at address 04000000h ~~000h~~.

Claim 147 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for accessing a predetermined storage ~~16-bit memory~~ location, wherein

bit positions 0-1 of the storage ~~memory~~ location specify one of four background priority levels;

bit positions 2-3 of the storage ~~memory~~ location specify a character base block value;

bit position 6 of the storage ~~memory~~ location specifies a mosaic enable/disable flag;

bit position 7 of the storage ~~memory~~ location selects between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

bit positions 8-12 of the storage ~~memory~~ location specify a screen base block; and

bit positions 14-15 of the storage ~~memory~~ location specify a screen size.

Claim 148 (Currently Amended): The memory of claim 147, wherein the predetermined storage ~~memory~~ location is at one or the other of address 04000008h ~~008h~~ and address 0400000Ah ~~00Ah~~.

Claim 149 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for accessing a predetermined storage ~~16-bit memory~~ location, wherein

bit positions 0-3 of the storage ~~memory~~ location specify a background character mosaic horizontal size;

bit positions 4-7 of the storage ~~memory~~ location specify a background character mosaic vertical size;

bit positions 8-11 of the storage ~~memory~~ location specify a moving object character mosaic horizontal size; and

bit positions 12-15 of the storage ~~memory~~ location specify a moving object character mosaic vertical size,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character.

Claim 150 (Currently Amended): The memory of claim 149, wherein the predetermined storage ~~memory~~ location is at address 0400004Ch ~~04Ch~~.

Claim 151 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for controlling rotation and/or scaling, the at least one further instruction writing rotation/scaling data to at least one storage location ~~register~~ including:

a 12-bit value specifying an x-coordinate reference starting point;

a 12-bit value specifying a y-coordinate reference starting point;

a 16-bit value specifying a distance of movement in the x-direction; and

a 16-bit value specifying a distance of movement in the y-direction.

Claim 152 (Currently Amended): The memory of claim 140, wherein the memory stores at least one further instruction for accessing a predetermined storage ~~16-bit memory~~ location, wherein

bit positions 0-4 of the storage ~~memory~~ location specify whether to display any or all of four backgrounds and a moving object in a first display window;

bit position 5 of the storage ~~memory~~ location specifies whether to enable color special effects within the first display window;

bit positions 8-12 of the storage ~~memory~~ location specify whether to display any or all of the four backgrounds and the moving object within a second display window different than the first display window; and

bit position 13 of the storage ~~memory~~ location specifies whether to enable color special effects within the second display window.

Claims 153-165 (Canceled).

Claim 166 (Currently Amended): A memory for use with a hand-held system for playing video games by displaying graphical information based at least in part on user interaction provided through operation of at least one user-manipulable control, the hand-held system including an object attribute memory (OAM) in a memory space from 07000000h to 070003FFh ~~storage~~, ~~a~~ color palette ~~random access memory~~ storage in a memory space from 05000000h to 050003FFh and ~~a~~ video ~~random access memory~~ storage in a memory space from 06000000h to 06017FFFh, the memory storing executable instructions for:

(a) storing no more than ~~writing up to~~ 128 different 48-bit moving object definitions in ~~to~~ the OAM at any one time ~~object attribute memory (OAM) storage~~;

(b) storing no more than ~~writing up to~~ 512 different 15-bit color values in ~~to~~ the color palette ~~random access memory~~ storage at any one time;

(c) writing ~~at least one of~~ moving object data and background data to the video ~~random access memory~~ storage; and

(d) generating a video game display that is responsive to the user-manipulable control and is based at least in part on the contents of the OAM ~~object attribute memory storage~~, the color palette ~~random access memory~~ storage, and the video ~~random access memory~~ storage,

wherein the memory stores further instructions that access two allocated frame buffers in the video ~~random access memory~~ storage to provide full motion video.

Claim 167 (Currently Amended): A memory for use with a hand-held system for playing video games by displaying graphical information based at least in part on user interaction provided through operation of at least one user-manipulable control, the hand-held system including an object attribute memory (OAM) in a memory space from 07000000h to 070003FFh ~~storage~~, ~~a~~ color palette ~~random access memory~~ storage in a memory space from 05000000h to 050003FFh and ~~a~~ video ~~random access memory~~ storage in a memory space from 06000000h to 06017FFFh, the memory storing executable instructions for:

(a) storing no more than ~~writing up to~~ 128 different 48-bit moving object definitions in ~~to~~ the OAM at any one time ~~object attribute memory (OAM) storage~~;

(b) storing no more than ~~writing up to~~ 512 different 15-bit color values in ~~to~~ the color palette ~~random access memory~~ storage at any one time;

(c) writing ~~at least one of~~ moving object data and background data to the video ~~random access memory~~ storage; and

(d) generating a video game display that is responsive to the user-manipulable control and is based at least in part on the contents of the OAM ~~object attribute memory storage~~, the color palette ~~random access memory~~ storage, and the video ~~random access memory~~ storage,

wherein the memory stores further instructions that control ~~the~~ alpha blending of plural display windows ~~independently~~.

Claim 168 (Currently Amended): A memory for use with a hand-held system for playing video games by displaying graphical information based at least in part on user interaction provided through operation of at least one user-manipulable control, the hand-held system including an object attribute memory (OAM) in a memory space from 07000000h to 070003FFh ~~storage~~, ~~a~~ color palette ~~random access memory~~ storage in a memory space from 05000000h to 050003FFh and ~~a~~ video ~~random access memory~~ storage in a memory space from 06000000h to 06017FFFh, the memory storing executable instructions for:

(a) storing no more than ~~writing up to~~ 128 different 48-bit moving object definitions in ~~to~~ the OAM at any one time ~~object attribute memory (OAM) storage~~;

(b) storing no more than ~~writing up to~~ 512 different 15-bit color values in ~~to~~ the color palette ~~random access memory~~ storage at any one time;

(c) writing ~~at least one of~~ moving object data and background data to the video ~~random access memory~~ storage; and

(d) generating a video game display that is responsive to the user-manipulable control and is based at least in part on the contents of the OAM ~~object attribute memory storage~~, the color palette ~~random access memory~~ storage, and the video ~~random access memory~~ storage,

wherein the memory stores further instructions that control ~~the~~ fade-in/fade-out of plural display windows ~~independently~~.

Claim 169 (Currently Amended): A memory for use with a hand-held system for playing video games by displaying graphical information based at least in part on user interaction provided through operation of at least one user-manipulable control, the hand-held system including an object attribute memory (OAM) in a memory space from 07000000h to 070003FFh ~~storage~~, ~~a~~ color palette ~~random access memory~~ storage in a memory space from 05000000h to 050003FFh and ~~a~~ video ~~random access memory~~ storage in a memory space from 06000000h to 06017FFFh, the memory storing executable instructions for:

(a) storing no more than ~~writing up to~~ 128 different 48-bit moving object definitions in ~~to~~ the OAM at any one time ~~object attribute memory (OAM) storage~~;

(b) storing no more than ~~writing up to~~ 512 different 15-bit color values in ~~to~~ the color palette ~~random access memory~~ storage at any one time;

(c) writing ~~at least one of~~ moving object data and background data to the video ~~random access memory~~ storage; and

(d) generating a video game display that is responsive to the user-manipulable control and is based at least in part on the contents of the OAM ~~object attribute memory storage~~, the color palette ~~random access memory~~ storage, and the video ~~random access memory~~ storage,

wherein the memory stores further instructions that control performance of arithmetic operations on two selected surfaces and processing for up to 16 levels of semi-transparency.

Claim 170 (Canceled).

Claim 171 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit position 0 of the storage ~~memory~~ location specifies vertical blanking interval status;

bit position 1 of the storage ~~memory~~ location specifies horizontal blanking interval status;

bit position 2 of the storage ~~memory~~ location specifies vertical counter matching or non-matching;

bit position 3 of the storage ~~memory~~ location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage ~~memory~~ location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage ~~memory~~ location specifies whether vertical counter matching interrupts are enabled,

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 172 (Currently Amended): The emulator ~~memory~~ of claim 171, wherein the predetermined storage ~~memory~~ location of the second system is at address 04000004h ~~004h~~.

Claim 173 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~memory~~ location of the second system, wherein

bit position 3 of the storage ~~memory~~ location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage ~~memory~~ location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage ~~memory~~ location specifies whether vertical counter matching interrupts are enabled,

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 174 (Currently Amended): The emulator ~~memory~~ of claim 173, wherein the predetermined storage ~~memory~~ location of the second system is at address 04000004h ~~004h~~.

Claim 175 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-2 of the storage ~~memory~~ location specify a background mode;

bit position 4 of the storage ~~memory~~ location specifies a display frame selector for selecting between two different frame buffers;

bit position 5 of the storage ~~memory~~ location specifies whether to render objects during horizontal blanking intervals;

bit position 6 of the storage ~~memory~~ location specifies a control bit for selecting between one-dimensional and two-dimensional object character mapping;

bit positions 8-12 of the storage ~~memory~~ location specify display of four different background screens and display of moving objects;

bit positions 13-14 of the storage ~~memory~~ location select display of two different windows; and

bit position 15 of the storage ~~memory~~ location selects display of an object window,

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 176 (Currently Amended): The emulator ~~memory~~ of claim 175, wherein the predetermined storage ~~memory~~ location of the second system is at address 04000000h ~~0000h~~.

Claim 177 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~memory~~ location of the second system, wherein

bit positions 0-1 of the storage ~~memory~~ location specify one of four background priority levels;

bit positions 2-3 of the storage ~~memory~~ location specify a character base block value;

bit position 6 of the storage ~~memory~~ location specifies a mosaic enable/disable flag;

bit position 7 of the storage ~~memory~~ location selects between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

bit positions 8-12 of the storage ~~memory~~ location specify a screen base block; and

bit positions 14-15 of the storage ~~memory~~ location specify a screen size,

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 178 (Currently Amended): The emulator ~~memory~~ of claim 177, wherein the predetermined storage ~~memory~~ location of the second system is at one or the other of address 04000008h ~~008h~~ and address 0400000Ah ~~00Ah~~.

Claim 179 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-3 of the storage ~~memory~~ location specify a background character mosaic horizontal size;

bit positions 4-7 of the storage ~~memory~~ location specify a background character mosaic vertical size;

bit positions 8-11 of the storage ~~memory~~ location specify a moving object character mosaic horizontal size; and

bit positions 12-15 of the storage ~~memory~~ location specify a moving object character mosaic vertical size,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 180 (Currently Amended): The emulator ~~memory~~ of claim 179, wherein the predetermined storage ~~memory~~ location of the second system is at address 0400004Ch ~~04Ch~~.

Claim 181 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for controlling rotation and/or scaling, the at least one further instruction writing rotation/scaling data to at least one ~~register~~ storage location of the second system including:

a 12-bit value specifying an x-coordinate reference starting point;

a 12-bit value specifying a y-coordinate reference starting point;

a 16-bit value specifying a distance of movement in the x-direction; and

a 16-bit value specifying a distance of movement in the y-direction, wherein the emulator emulates the storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated storage location.

Claim 182 (Currently Amended): The emulator ~~memory~~ of claim 201 ~~170~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-4 of the storage ~~memory~~ location specify whether to display any or all of four backgrounds and a moving object in a first display window;

bit position 5 of the storage ~~memory~~ location specifies whether to enable color special effects within the first display window;

bit positions 8-12 of the storage ~~memory~~ location specify whether to display any or all of the four backgrounds and the moving object within a second display window different than the first display window; ~~and~~

bit position 13 of the storage ~~memory~~ location specifies whether to enable color special effects within the second display window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 183 (Canceled).

Claim 184 (Currently Amended): The emulator ~~memory~~ of claim 204 ~~183~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit position 0 of the storage ~~memory~~ location specifies vertical blanking interval status;

bit position 1 of the storage ~~memory~~ location specifies horizontal blanking interval status;

bit position 2 of the storage ~~memory~~ location specifies vertical counter matching or non-matching;

bit position 3 of the storage ~~memory~~ location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage ~~memory~~ location specifies whether horizontal blanking interval interrupts are enabled; ~~and~~

bit position 5 of the storage ~~memory~~ location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 185 (Currently Amended): The emulator ~~memory~~ of claim 184, wherein the predetermined storage ~~memory~~ location of the second system is at address 04000004h ~~004h~~.

Claim 186 (Currently Amended): The emulator ~~memory~~ of claim 204 ~~183~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit position 3 of the storage ~~memory~~ location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage ~~memory~~ location specifies whether horizontal blanking interval interrupts are enabled; ~~and~~

bit position 5 of the storage ~~memory~~ location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 187 (Currently Amended): The emulator ~~memory~~ of claim 186, wherein the predetermined storage ~~memory~~ location of the second system is at address 04000004h ~~004h~~.

Claim 188 (Currently Amended): The emulator ~~memory~~ of claim 204 ~~183~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-2 of the storage ~~memory~~ location specify a background mode;

bit position 4 of the storage ~~memory~~ location specifies a display frame selector for selecting between two different frame buffers;

bit position 5 of the storage ~~memory~~ location specifies whether to render objects during horizontal blanking intervals;

bit position 6 of the storage ~~memory~~ location specifies a control bit for selecting between one-dimensional and two-dimensional object character mapping;

bit positions 8-12 of the storage ~~memory~~ location specify display of four different background screens and display of moving objects;

bit positions 13-14 of the storage ~~memory~~ location select display of two different windows; and

bit position 15 of the storage ~~memory~~ location selects display of an object window, and wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 189 (Currently Amended): The emulator ~~memory~~ of claim 188, wherein the predetermined storage ~~memory~~ location of the second system is at address 04000000h ~~000h~~.

Claim 190 (Currently Amended): The emulator ~~memory~~ of claim 204 ~~183~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-1 of the storage ~~memory~~ location specify one of four background priority levels;

bit positions 2-3 of the storage ~~memory~~ location specify a character base block value;

bit position 6 of the storage ~~memory~~ location specifies a mosaic enable/disable flag;

bit position 7 of the storage ~~memory~~ location selects between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

bit positions 8-12 of the storage ~~memory~~ location specify a screen base block; and

bit positions 14-15 of the storage ~~memory~~ location specify a screen size, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 191 (Currently Amended): The emulator ~~memory~~ of claim 190, wherein the predetermined storage ~~memory~~ location of the second system is at one or the other of address 04000008h ~~008h~~ and address 0400000Ah ~~00Ah~~.

Claim 192 (Currently Amended): The emulator ~~memory~~ of claim 204 ~~183~~, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-3 of the storage ~~memory~~ location specify a background character mosaic horizontal size;

bit positions 4-7 of the storage ~~memory~~ location specify a background character mosaic vertical size;

bit positions 8-11 of the storage ~~memory~~ location specify a moving object character mosaic horizontal size; and

bit positions 12-15 of the storage ~~memory~~ location specify a moving object character mosaic vertical size,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 193 (Currently Amended): The emulator ~~memory~~ of claim 192, wherein the predetermined storage ~~memory~~ location of the second system is at address 0400004Ch ~~04Ch~~.

Claim 194 (Currently Amended): The emulator ~~memory~~ of claim 204 ~~183~~, wherein the memory stores at least one further instruction executable by the second processor for controlling rotation and/or scaling, the at least one further instruction writes rotation/scaling data to at least one ~~register~~ storage location of the second system including:

a 12-bit value specifying an x-coordinate reference starting point;

a 12-bit value specifying a y-coordinate reference starting point;

a 16-bit value specifying a distance of movement in the x-direction; and

a 16-bit value specifying a distance of movement in the y-direction, and

wherein the emulator emulates the storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated storage location.

Claim 195 (Currently Amended): The emulator ~~memory~~ of claim 183, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage ~~16-bit memory~~ location of the second system, wherein

bit positions 0-4 of the storage ~~memory~~ location specify whether to display any or all of four backgrounds and a moving object in a first display window;

bit position 5 of the storage ~~memory~~ location specifies whether to enable color special effects within the first display window;

bit positions 8-12 of the storage ~~memory~~ location specify whether to display any or all of the four backgrounds and the moving object within a second display window different than the first display window; and

bit position 13 of the storage ~~memory~~ location specifies whether to enable color special effects within the second display window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claims 196-200 (Canceled).

Claim 201 (New): An emulator for execution on a first system including a first processor, first user controls and a first display, the emulator at least in part emulating the operation of a hand-held second system used to play video games, the second system comprising a second processor, second user controls, a second display, an object attribute memory (OAM) in a memory space from 07000000h to 070003FFh, color palette storage in a memory space from 050000000h to 050003FFh and video storage in a memory space from 060000000h to 06017FFFh, the emulator comprising:

first instructions for execution by the first system to access a memory storing video game program instructions, wherein at least some of the video game program instructions stored in the memory are incompatible with the first system and wherein the video game program instructions are executable by the second system to:

(a) store no more than 128 different 48-bit moving object definitions in the OAM of the second system at any one time;

(b) store no more than 512 different 15-bit color values in the color palette storage of the second system at any one time;

(c) write moving object data and background data to the video storage of the second system; and

(d) generate a video game display on the second display based at least in part on inputs to the second user controls and on the contents of the OAM, the color palette storage, and the video storage,

wherein the video game program instructions cause background data to be selectively written to the video storage as either character data or bitmap data so that backgrounds of the video game display are selectively rendered in a character mode or in a bitmap mode;

second instructions for execution by the first system to transform at least some of the video game program instructions stored in the memory that are incompatible with the first system into transformed instructions that are compatible with the first system; and

third instructions for execution by the first system to use the transformed video game program instructions in generating a video game display on the first display.

Claim 202 (New): The emulator of claim 201, wherein the first system comprises a hand-held system running a Palm operating system.

Claim 203 (New): The emulator of claim 201, wherein the first system comprises a hand-held system running a Windows operating system.

Claim 204 (New): An emulator for execution on a first system including a first processor, first user controls and a first display, the emulator at least in part emulating the operation of a hand-held second system used to play video games, the second system comprising a second processor, second user controls, a second display, an object attribute memory (OAM) storage in a memory space from 07000000h to 070003FFh, color palette storage in a memory space from 050000000h to 050003FFh and video storage in a memory space from 060000000h to 06017FFFh, the emulator comprising:

first instructions for execution by the first system to access a memory storing video game program instructions, wherein at least some of the video game program instructions stored in the memory are incompatible with the first system and wherein the video game program instructions are executable by the second system to:

(a) store no more than 128 different 48-bit moving object definitions in the OAM of the second system at any one time;

(b) store no more than 512 different 15-bit color values in the color palette storage of the second system at any one time;

(c) write moving object data and background data to the video storage of the second system; and

(d) generate a video game display on the second display based at least in part on inputs to the second user controls and on the contents of the OAM, the color palette storage, and the video storage,

wherein the video game program instructions access two allocated frame buffers in the video storage to provide full motion video;

second instructions for execution by the first system to transform at least some of the video game program instructions stored in the memory that are incompatible with the first system into transformed instructions that are compatible with the first system; and

third instructions for execution by the first system to use the transformed video game program instructions to generate a video game display on the first display.

Claim 205 (New): The emulator of claim 204, wherein the first system comprises a hand-held system running a Palm operating system.

Claim 206 (New): The emulator of claim 204, wherein the first system comprises a hand-held system running a Windows operating system.

Claim 207 (New): An emulator for execution on a first system including a first processor, first user controls and a first display, the emulator at least in part emulating the operation of a hand-held second system used to play video games, the second system comprising a second processor, second user controls, a second display, an object attribute memory (OAM) storage in a memory space from 07000000h to 070003FFh, color palette storage in a memory space from 050000000h to 050003FFh and video storage in a memory space from 060000000h to 06017FFFh, the emulator comprising:

first instructions for execution by the first system to access a memory storing video game program instructions, wherein at least some of the video game program instructions stored in the memory are incompatible with the first system and wherein the video game program instructions are executable by the second system to:

(a) store no more than 128 different 48-bit moving object definitions in the OAM of the second system at any one time;

(b) store no more than 512 different 15-bit color values in the color palette storage of the second system at any one time;

(c) write moving object data and background data to the video storage of the second system; and

(d) generate a video game display on the second display based at least on part on inputs to the second user controls and on the contents of the OAM, the color palette storage, and the video storage,

wherein the video game program instructions control alpha blending of plural display windows;

second instructions for execution by the first system to transform at least some of the video game program instructions stored in the memory that are incompatible with the first system into transformed instructions that are compatible with the first system; and

third instructions for execution by the first system to use the transformed video game program instructions to generate a video game display on the first display.

Claim 208 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit position 0 of the storage location specifies vertical blanking interval status;

bit position 1 of the storage location specifies horizontal blanking interval status;

bit position 2 of the storage location specifies vertical counter matching or non-matching;

bit position 3 of the storage location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage location specifies whether horizontal blanking interval interrupts are enabled; and bit position 5 of the storage location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 209 (New): The emulator of claim 208, wherein the predetermined storage location of the second system is at address 04000004h.

Claim 210 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit position 3 of the storage location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 211 (New): The emulator of claim 210, wherein the predetermined storage location of the second system is at address 04000004h.

Claim 212 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-2 of the storage location specify a background mode;

bit position 4 of the storage location specifies a display frame selector for selecting between two different frame buffers;

bit position 5 of the storage location specifies whether to render objects during horizontal blanking intervals;

bit position 6 of the storage location specifies a control bit for selecting between one-dimensional and two-dimensional object character mapping;

bit positions 8-12 of the storage location specify display of four different background screens and display of moving objects;

bit positions 13-14 of the storage location select display of two different windows; and

bit position 15 of the storage location selects display of an object window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 213 (New): The emulator of claim 212, wherein the predetermined storage location of the second system is at address 04000000h.

Claim 214 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-1 of the storage location specify one of four background priority levels;

bit positions 2-3 of the storage location specify a character base block value;

bit position 6 of the storage location specifies a mosaic enable/disable flag;

bit position 7 of the storage location selects between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

bit positions 8-12 of the storage location specify a screen base block; and

bit positions 14-15 of the storage location specify a screen size, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 215 (New): The emulator of claim 214, wherein the predetermined storage location of the second system is at one or the other of address 04000008h and address 0400000Ah.

Claim 216 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-3 of the storage location specify a background character mosaic horizontal size;

bit positions 4-7 of the storage location specify a background character mosaic vertical size;

bit positions 8-11 of the storage location specify a moving object character mosaic horizontal size; and

bit positions 12-15 of the storage location specify a moving object character mosaic vertical size,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 217 (New): The emulator of claim 216, wherein the predetermined storage location of the second system is at address 0400004Ch.

Claim 218 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for controlling rotation and/or scaling, the at least one further instruction writes rotation/scaling data to at least one storage location of the second system including:

a 12-bit value specifying an x-coordinate reference starting point;

a 12-bit value specifying a y-coordinate reference starting point;

a 16-bit value specifying a distance of movement in the x-direction; and

a 16-bit value specifying a distance of movement in the y-direction, and

wherein the emulator emulates the storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated storage location.

Claim 219 (New): The emulator of claim 207, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-4 of the storage location specify whether to display any or all of four backgrounds and a moving object in a first display window;

bit position 5 of the storage location specifies whether to enable color special effects within the first display window;

bit positions 8-12 of the storage location specify whether to display any or all of the four backgrounds and the moving object within a second display window different than the first display window; and

bit position 13 of the storage location specifies whether to enable color special effects within the second display window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 220 (New): The emulator of claim 207, wherein the first system comprises a hand-held system running a Palm operating system.

Claim 221 (New): The emulator of claim 207, wherein the first system comprises a hand-held system running a Windows operating system.

Claim 222 (New): An emulator for execution on a first system including a first processor, first user controls and a first display, the emulator at least in part emulating the operation of a hand-held second system used to play video games, the second system comprising a second processor, second user controls, a second display, an object attribute memory (OAM) storage in a memory space from 07000000h to 070003FFh, color palette storage in a memory space from 050000000h to 050003FFh and video storage in a memory space from 060000000h to 06017FFFh, the emulator comprising:

first instructions for execution by the first system to access a memory storing video game program instructions, wherein at least some of the video game program instructions stored in the memory are incompatible with the first system and wherein the video game program instructions are executable by the second system to:

(a) storeno more than 128 different 48-bit moving object definitions in the OAM of the second system at any one time;

(b) store no more than 512 different 15-bit color values in the color palette storage of the second system at any one time;

(c) write moving object data and background data to the video storage of the second system; and

(d) generate a video game display on the second display based at least in part on inputs to the second user controls and on the contents of the OAM, the color palette storage, and the video storage,

wherein the video game program instructions control fade-in/fade-out of plural display windows;

second instructions for execution by the first system to transform at least some of the video game program instructions stored in the memory that are incompatible with the first system into transformed instructions that are compatible with the first system; and

third instructions for execution by the first system to use the transformed video game program instructions to generate a video game display on the first display.

Claim 223 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit position 0 of the storage location specifies vertical blanking interval status;

bit position 1 of the storage location specifies horizontal blanking interval status;

bit position 2 of the storage location specifies vertical counter matching or non-matching;

bit position 3 of the storage location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 224 (New): The emulator of claim 223, wherein the predetermined storage location of the second system is at address 04000004h.

Claim 225 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit position 3 of the storage location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 226 (New): The emulator of claim 225, wherein the predetermined storage location of the second system is at address 04000004h.

Claim 227 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-2 of the storage location specify a background mode;

bit position 4 of the storage location specifies a display frame selector for selecting between two different frame buffers;

bit position 5 of the storage location specifies whether to render objects during horizontal blanking intervals;

bit position 6 of the storage location specifies a control bit for selecting between one-dimensional and two-dimensional object character mapping;

bit positions 8-12 of the storage location specify display of four different background screens and display of moving objects;

bit positions 13-14 of the storage location select display of two different windows; and

bit position 15 of the storage location selects display of an object window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 228 (New): The emulator of claim 227, wherein the predetermined storage location of the second system is at address 04000000h.

Claim 229 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-1 of the storage location specify one of four background priority levels;

bit positions 2-3 of the storage location specify a character base block value;

bit position 6 of the storage location specifies a mosaic enable/disable flag;

bit position 7 of the storage location selects between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

bit positions 8-12 of the storage location specify a screen base block; and

bit positions 14-15 of the storage location specify a screen size, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 230 (New): The emulator of claim 229, wherein the predetermined storage location of the second system is at one or the other of address 04000008h and address 0400000Ah.

Claim 231 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-3 of the storage location specify a background character mosaic horizontal size;

bit positions 4-7 of the storage location specify a background character mosaic vertical size;

bit positions 8-11 of the storage location specify a moving object character mosaic horizontal size; and

bit positions 12-15 of the storage location specify a moving object character mosaic vertical size,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 232 (New): The emulator of claim 231, wherein the predetermined storage location of the second system is at address 0400004Ch.

Claim 233 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for controlling rotation and/or scaling, the at least one further instruction writes rotation/scaling data to at least one storage location of the second system including:

a 12-bit value specifying an x-coordinate reference starting point;

a 12-bit value specifying a y-coordinate reference starting point;

a 16-bit value specifying a distance of movement in the x-direction; and

a 16-bit value specifying a distance of movement in the y-direction, and

wherein the emulator emulates the storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated storage location.

Claim 234 (New): The emulator of claim 222, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-4 of the storage location specify whether to display any or all of four backgrounds and a moving object in a first display window;

bit position 5 of the storage location specifies whether to enable color special effects within the first display window;

bit positions 8-12 of the storage location specify whether to display any or all of the four backgrounds and the moving object within a second display window different than the first display window; and

bit position 13 of the storage location specifies whether to enable color special effects within the second display window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 235 (New): The emulator of claim 222, wherein the first system comprises a hand-held system running a Palm operating system.

Claim 236 (New): The emulator of claim 222, wherein the first system comprises a hand-held system running a Windows operating system.

Claim 237 (New): An emulator for execution on a first system including a first processor, first user controls and a first display, the emulator at least in part emulating the operation of a hand-held second system used to play video games, the second system comprising a second processor, second user controls, a second display, an object attribute memory (OAM) storage in a memory space from 07000000h to 070003FFh, color palette storage in a memory space from 050000000h to 050003FFh and video storage in a memory space from 060000000h to 06017FFFh, the emulator comprising:

first instructions for execution by the first system to access a memory storing video game program instructions, wherein at least some of the video game program instructions stored in the memory are incompatible with the first system and wherein the video game program instructions are executable by the second system to:

(a) store no more than 128 different 48-bit moving object definitions in the OAM of the second system at any one time;

(b) store no more than 512 different 15-bit color values in the color palette storage of the second system at any one time;

(c) write moving object data and background data to the video storage of the second system; and

(d) generate a video game display on the second display based at least in part on inputs the second user controls and on the contents of the OAM, the color palette storage, and the video storage,

wherein the video game program instructions control performance of arithmetic operations on two selected surfaces and processing for up to 16 levels of semi-transparency;

second instructions for execution by the first system to transform at least some of the video game program instructions stored in the memory that are incompatible with the first system into transformed instructions that are compatible with the first system; and

third instructions for execution by the first system to use the transformed video game program instructions to generate a video game display on the first display.

Claim 238 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit position 0 of the storage location specifies vertical blanking interval status;

bit position 1 of the storage location specifies horizontal blanking interval status;

bit position 2 of the storage location specifies vertical counter matching or non-matching;

bit position 3 of the storage location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 239 (New): The emulator of claim 238, wherein the predetermined storage location of the second system is at address 04000004h.

Claim 240 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit position 3 of the storage location specifies whether vertical blanking interval interrupts are enabled;

bit position 4 of the storage location specifies whether horizontal blanking interval interrupts are enabled; and

bit position 5 of the storage location specifies whether vertical counter matching interrupts are enabled, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 241 (New): The emulator of claim 240, wherein the predetermined storage location of the second system is at address 04000004h.

Claim 242 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-2 of the storage location specify a background mode;

bit position 4 of the storage location specifies a display frame selector for selecting between two different frame buffers;

bit position 5 of the storage location specifies whether to render objects during horizontal blanking intervals;

bit position 6 of the storage location specifies a control bit for selecting between one-dimensional and two-dimensional object character mapping;

bit positions 8-12 of the storage location specify display of four different background screens and display of moving objects;

bit positions 13-14 of the storage location select display of two different windows; and

bit position 15 of the storage location selects display of an object window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 243 (New): The emulator of claim 242, wherein the predetermined storage location of the second system is at address 04000000h.

Claim 244 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-1 of the storage location specify one of four background priority levels;

bit positions 2-3 of the storage location specify a character base block value;

bit position 6 of the storage location specifies a mosaic enable/disable flag;

bit position 7 of the storage location selects between a 16 color, 16 palette color mode and a 256 color, one palette color mode;

bit positions 8-12 of the storage location specify a screen base block; and

bit positions 14-15 of the storage location specify a screen size, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 245 (New): The emulator of claim 244, wherein the predetermined storage location of the second system is at one or the other of address 04000008h and address 0400000Ah.

Claim 246 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-3 of the storage location specify a background character mosaic horizontal size;

bit positions 4-7 of the storage location specify a background character mosaic vertical size;

bit positions 8-11 of the storage location specify a moving object character mosaic horizontal size; and

bit positions 12-15 of the storage location specify a moving object character mosaic vertical size,

wherein the mosaic sizes specify how many dots in an original character should be replaced by a virtual character, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 247 (New): The emulator of claim 246, wherein the predetermined storage location of the second system is at address 0400004Ch.

Claim 248 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for controlling rotation and/or scaling, the at least one further instruction writes rotation/scaling data to at least one storage location of the second system including:

a 12-bit value specifying an x-coordinate reference starting point;

a 12-bit value specifying a y-coordinate reference starting point;

a 16-bit value specifying a distance of movement in the x-direction; and

a 16-bit value specifying a distance of movement in the y-direction, and

wherein the emulator emulates the storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated storage location.

Claim 249 (New): The emulator of claim 237, wherein the memory stores at least one further instruction executable by the second processor for accessing a predetermined storage location of the second system, wherein

bit positions 0-4 of the storage location specify whether to display any or all of four backgrounds and a moving object in a first display window;

bit position 5 of the storage location specifies whether to enable color special effects within the first display window;

bit positions 8-12 of the storage location specify whether to display any or all of the four backgrounds and the moving object within a second display window different than the first display window; and

bit position 13 of the storage location specifies whether to enable color special effects within the second display window, and

wherein the emulator emulates the predetermined storage location and transforms the at least one further instruction to at least one instruction executable by the first processor to access the emulated predetermined storage location.

Claim 250 (New): The emulator of claim 237, wherein the first system comprises a hand-held system running a Palm operating system.

Claim 251 (New): The emulator of claim 237, wherein the first system comprises a hand-held system running a Windows operating system.

Amendment to the Title:

Please change the title to:

MEMORY FOR VIDEO GAME SYSTEM AND EMULATOR USING THE MEMORY